Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1997 Proceedings

Americas Conference on Information Systems (AMCIS)

8-15-1997

What Do I-CASE Tool-Users Think About the Toolsets?

Prachit Hawat
The University of Calgary

What Do I-CASE Tool-Users Think About the Toolsets?

Prachit Hawat
The University of Calgary, Canada, and
Chulalongkorn University, Thailand

Introduction

An integrated CASE toolset is a set of computer programs that automates and integrates most of the tasks in the systems development life cycle. While the project progresses through the stages of systems development, the toolset creates a knowledge base (CASE encyclopedia) of the organization, its goals, strategies, and business rules as well as data models and other systems development-related information [Martin, 1989] so that all the team members could share. The technology represents the long-time- ultimate-goal of CASE industry to integrate individual CASE tools [Pressman, 1992; Martin, 1989] and is hence expected to render maximum software development productivity and software quality.

While little research has been done on the use of I- CASE toolsets, there is also a paucity of research on the teamwork aspects of such a use. Such an issue is particularly important to the use of I-CASE since the toolsets are designed to be used by large development project teams. In addition, the information stored on CASE encyclopedia needs to be created and shared by every team member and in every phase of software development. However, a review of the literature to date has uncovered only one study on the effects of the introduction of CASE tools on social relations among project team members. Orlikowski [1989] examined this issue at a large software consulting firm, Beta, which introduced a number of individual CASE tools developed in-house by technical personnel. The introduction resulted in the inclusion of technical personnel, who had played "purely support role" for the functional personnel who used the tools to develop applications for their clients. Not only did the technical personnel build and install CASE tools, but also were more involved in analysis and design decisions due to the constraints inherent in the tools. Thus, they became the central group in systems development tasks. Consequently, this situation caused "territorialism, resentment, and rebellion" on the part of the functional project team members. Some of them declined "to conform to the tools and the "team" way of doing things" (p. 209). The author called for further research to determine the condition that would affect such a tension between technical and functional project team members.

The purpose of this paper is to gain further understanding of the relationship between the use of I- CASE and teamwork in software development.

Method

The discussion in this paper draws on a study that looked into the variation in the use of I-CASE at both organizational and individual levels. Since this study seeks to explore the perception of and experience with a relatively new technology, a qualitative research approach was chosen. Nineteen semi-structured and unstructured interviews with seventeen IS developers with experience with commercially available information engineering (IE)- based I-CASE toolsets, ADW and IEF, were conducted during November 1995 to February 1997. The individual interviews varied in length from approximately 20 minutes to one hour. The informants included programmer analysts, project leaders, a consultant, a programming supervisor, a data administrator, and I-CASE technical support persons from in-house IS shop in six organizations across a range of industries in the U.S. and Canada. While these informants had been working in IS 10-25 years, their experience with I-CASE toolsets varied widely, from 6 months to 8 years.

Findings

This section is organized as follows. First, the contradiction between the findings in this study and those in Orlikowski's will be discussed. Second, the perception of the tool-users of the effects of the toolset on their teamwork will be described. Finally, some requisites for use of the toolsets as pointed out by the informants in this study are reported. In contrast to the previous study reviewed above, in this study there was no disruption in social relations among project team members as a result of introduction of an I- CASE toolset. From the information available reported in Orlikowski's study [1989], there are some differences between the context of the organizations under this study and that of Beta that could contribute to the contradiction in the findings. These differences include (1) who developed CASE tools, and (2) the positioning of technical support in system development project team. As mentioned earlier, CASE tools at Beta were developed by technical personnel. Having realized this fact, functional personnel who used the tools perceived that their work depended on the technical team. In addition, they also rationalized that technical personnel ignored the deficiency of their tools. A functional person said, They are not open to criticism. They feel some ownership of the tools and so are very defensive. I guess that's human nature...they just don't want to know that their tools are defective or weak. (p. 203)

Technical support personnel at Beta were assigned to be part of the development team and thus become more involved in the development tasks, whereas those in this study are in separate unit that plays only support role. As a result, the technical personnel at Beta were perceived as having "stolen the show" from as well as having less or no concern for needs for support of the functional personnel.

Perceptions of the tool-users in this study

Systems developers in this study perceive the I-CASE toolset which they used as consisting of the tools and the information engineering methodology. The tools provide them with automated assistance as well as consistency check in various systems development tasks, such as modeling, diagramming, prototyping, and documentation. On

the other hand, IE methodology, like any other structured methodologies, represents a collection of concepts, principles, and methods, and thus specifies tasks and techniques in systems development. Nonetheless, there are three distinct characteristics of IE that enable IE-based I-CASE toolsets to enhance the efficiency and effectiveness of systems development teams.

First, IE is basically data-oriented systems analysis and design whereas most traditional structured methodologies are process-oriented. The distinction is that while a data-oriented approach seeks to capture data and their relationship, the process-oriented approach tries to automate business processes and activities. In addition, in IE, the terminology used was defined precisely and hence provides common understanding and usage. However, in a process-oriented view, most of the interpretation of such term as "business process" and the like are left to the systems developers to figure out on their own [Hirschheim et al., 1995]. Often how a developer defines certain terms is not congruent with those understood by their team members. An interviewee recalled, "everybody was used to doing things their own way"

Secondly, IE-based I-CASE toolsets compel systems developers to look at the big picture of which their work is a part and see how their work relates to those of others. An interviewee compared,

Before, the document unit of works we were looking at was one module, in the I-CASE you've got to look at a model. It's more than one screen, more than one job...that's good because it makes you look at things from a higher perspective as opposed to looking at everything in such a small scope. You have to look at how one screen interacts with another.

Not only does the toolset force its users to look for such information but the toolsets also provide the means to do that, as well as access other information. With a CASE encyclopedia, each of the development team members has access to the model and is able to check out the most up- to-date model to work on a personal workstations and check in the modified model back to the encyclopedia. The encyclopedia provides access control, auditing functions, and security check of all access and keeps the archive of all the previous versions of the model.

Thirdly, since I-CASE toolsets are integrated, they have capabilities, such as consistency and integrity check, to transfer deliverables smoothly from one task or one project to subsequent ones. This eliminates the problem of incompatibilities which would result in the team members having to retranslate and rework the deliverables. Requisites for effective use of I-CASE toolsets

After an I-CASE toolset is acquired, positive outcomes of the toolset do not occur automatically. There are still a number of requisites to be met in order for the use to be continuous and effective. Four factors frequently mentioned by the tool-users are: sufficient training in IE methodology and the toolset, communication among tool- users, guidelines and standards for the use of the toolset, and good team spirit to begin with.

a) Sufficient training in IE methodology and in the toolset

Most of the informants mentioned that having to develop a systems using the concept of IE as "a change in the mindset altogether." Moreover, some others, who had been mainframe application developers and hardly used PCs remembered that it was "quite overwhelming" to have to learn the methodology and learn to use Windows at the same time. Similarly, Reeh [1995] reported, "IEF was only regarded as difficult in the sense that it was different. It was different with regard to the underlying methodology and its `point and click' approach toward systems development" (p.137). Thus, sufficient training both in IE as well as in the toolset interface (Windows or OS/2 or some other type) is crucial to the use of the toolsets.

An important facet of such training is that it should be given prior to the use of the toolset with additional on-the- job training. The lack of sufficient training prior to use led to the abandonment of ADW toolset in an oil and gas company in this study. A system analyst who experienced the use of the toolset in 1991 explained that there was the belief in his company at that time that, "You could not train people to apply the methodology very well if you don't have the tool." They bought ADW toolset and used it to "generate reams of paper on different ways of analyzing ... but did not really know how to interpret all of these." They ended up falling back to the traditional method of development.

b) Guidelines and standards

Most of the IS shops in this study have some form of guidelines and/or standards for the tool-users to follow. Some informants argued that they need to have a clear idea upfront as to what they can to accomplish. For example, one informant pointed out, "CASE tool is like a fire, it's a useful tool but a terrible master." He cautioned that tool- users should not "go wild with it and just generate every possible report and every possible diagram." Another informant in a State government unit mentioned that in the past each individual tool-users had a unique way of developing using the toolset which resulted in a lot of problems. He also emphasized that merely following what is written in the manual that comes with the toolset as well as what is taught in the formal training session is not adequate to assure the smooth development in a large project. He said, "It was basically what you have to do inside the toolset, (such as) you move the mouse here, and you click ... It's what you are doing as opposed to how and why you are doing it."

In this study, such guidelines and standards vary from organization to organization. For example, a financial company in this study does not have a written standards document but has a person to dictate the sequence of the tasks that need to be done in any development project and which of these tasks would be assisted by the toolset or other software development tools. A government-related organization uses a combination of guidelines and standards such as using existing IBM GUI standard for GUI as well as using their in-house guidelines to specify the steps of development their deliverables.

c) Communication among IS staff

Communication among IS staff is important for the success of any development project. Nevertheless, communication seemed more important in the projects that use I-CASE. A technical support staff member for I- CASE users at a large financial organization in Chicago mentioned that because I-CASE is an integrated product, it takes a lot of coordination and communication. He gave an example of the need of communication in taking a data model that resides on the repository on the mainframe or a network-based encyclopedia.

d) Good team spirit to begin with

Learning to use the toolsets involves using it on the job. Most of informants mentioned that there is a long leaning curve associated with learning to use the toolsets. Many of them feel frustrated along this learning process which is partly due to the pressure to get the job done while still learning to use the toolset. They also mentioned that their colleagues helped them in the learning as well as providing emotional support for them while working together on the job. Even in the formal training, some help from colleagues also improves one's proficiency in using the toolset. A few informants mentioned that they also learned from being assigned to help train others in the formal training sessions. Interestingly, the strategy that many organizations use in training the tool-users on the job is to team up inexperienced tool-users with the experienced ones and/or with the outside consultants. Thus, good team spirit should be established and promoted prior to the introduction of the toolset.

Conclusion

I-CASE toolsets enhances effectiveness and efficiency of systems development project team through the use of their encyclopedia and other integrated features. Since using IE-based I-CASE toolsets involves a number of people and the understanding of IE methodology, it requires sufficient training in IE and the toolset, guidelines and standards, communication among the tool-users, and good team spirit.

References available upon request from author.